Assignment 4

ML-based prediction system to predict the career for a new graduate

First, I analyzed the data.

```
uploaded = files.upload()
records = pd.read_csv(io.BytesIO(uploaded['roo_data.csv']))
records.head()
```

```
Choose Files  roo_data.csv
• roo_data.csv(text/csv) - 4872181 bytes, last modified: 11/29/2022 - 100% done
Saving roo_data.csv to roo_data (2).csv
```

	Acedamic percentage in Operating Systems	percentage in Algorithms	Percentage in Programming Concepts	Percentage in Software Engineering	Percentage in Computer Networks	Percentage in Electronics Subjects	Percentage in Computer Architecture	Percentage in Mathematics	Percentage in Communication skills	Hours working per day	..
0	69	63	78	87	94	94	87	84	61	9	
1	78	62	73	60	71	70	73	84	91	12	
2	71	86	91	87	61	81	72	72	94	11	
3	76	87	60	84	89	73	62	88	69	7	
4	92	62	90	67	71	89	73	71	73	4	

5 rows × 39 columns

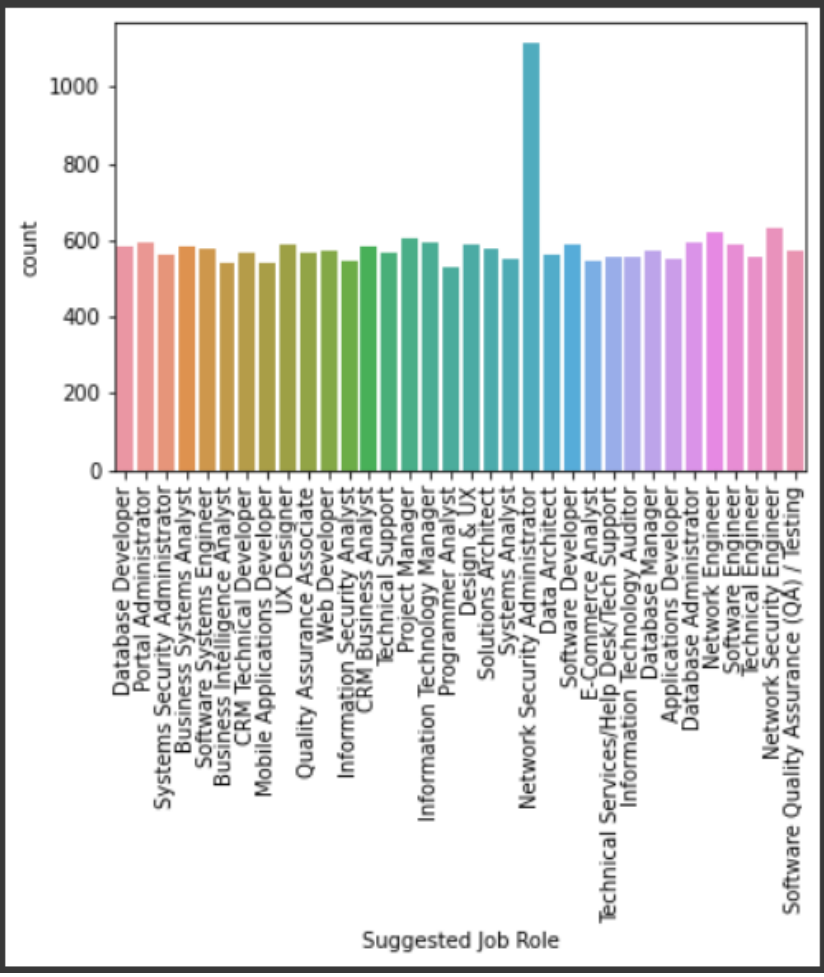
```
records.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 39 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Acedamic percentage in Operating Systems  20000 non-null  int64
 1   percentage in Algorithms              20000 non-null  int64
 2   Percentage in Programming Concepts    20000 non-null  int64
 3   Percentage in Software Engineering    20000 non-null  int64
 4   Percentage in Computer Networks       20000 non-null  int64
 5   Percentage in Electronics Subjects    20000 non-null  int64
 6   Percentage in Computer Architecture   20000 non-null  int64
 7   Percentage in Mathematics             20000 non-null  int64
 8   Percentage in Communication skills    20000 non-null  int64
 9   Hours working per day                 20000 non-null  int64
 10  Logical quotient rating               20000 non-null  int64
 11  hackathons                            20000 non-null  int64
 12  coding skills rating                  20000 non-null  int64
 13  public speaking points                20000 non-null  int64
 14  can work long time before system?     20000 non-null  object
 15  self-learning capability?             20000 non-null  object
 16  Extra-courses did                     20000 non-null  object
 17  certifications                        20000 non-null  object
 18  workshops                             20000 non-null  object
 19  talenttests taken?                    20000 non-null  object
 20  olympiads                             20000 non-null  object
 21  reading and writing skills            20000 non-null  object
```

```
[14] n = records.shape[1] - 1
     print("Unique values for " + records.columns[n] + "\n")
     print(records.iloc[:,n].unique())
```

```
Unique values for Suggested Job Role

['Database Developer' 'Portal Administrator'
 'Systems Security Administrator' 'Business Systems Analyst'
 'Software Systems Engineer' 'Business Intelligence Analyst'
 'CRM Technical Developer' 'Mobile Applications Developer' 'UX Designer'
 'Quality Assurance Associate' 'Web Developer'
 'Information Security Analyst' 'CRM Business Analyst' 'Technical Support'
 'Project Manager' 'Information Technology Manager' 'Programmer Analyst'
 'Design & UX' 'Solutions Architect' 'Systems Analyst'
 'Network Security Administrator' 'Data Architect' 'Software Developer'
 'E-Commerce Analyst' 'Technical Services/Help Desk/Tech Support'
 'Information Technology Auditor' 'Database Manager'
 'Applications Developer' 'Database Administrator' 'Network Engineer'
 'Software Engineer' 'Technical Engineer' 'Network Security Engineer'
 'Software Quality Assurance (QA) / Testing']
```

```
chart = sns.countplot(x=last1)
chart.set_xticklabels(chart.get_xticklabels(),rotation=90)
plt.show()
```



Splitting into training and testing:

```
rest_train, rest_test, last_train, last_test = tts(rest1, last1, test_size = 0.20)

tmp = mlpc(random_state = 40)

cls = tmp.fit(rest_train, last_train)

acs(cls.predict(rest_test), last_test)
```

Feeding into ANN using one-hot encoding:

```
rest = records.iloc[:,:-1]
last = records.iloc[:,-1]

tmp = ohe()

rest1 = tmp.fit_transform(rest)
last1 = last.copy(deep = True)
```

Outcomes:

```
/usr/local/lib/python3.7/dist-packages/sklearn/n
  ConvergenceWarning,
0.03275
```

Training and Testing accuracies in the starting are:

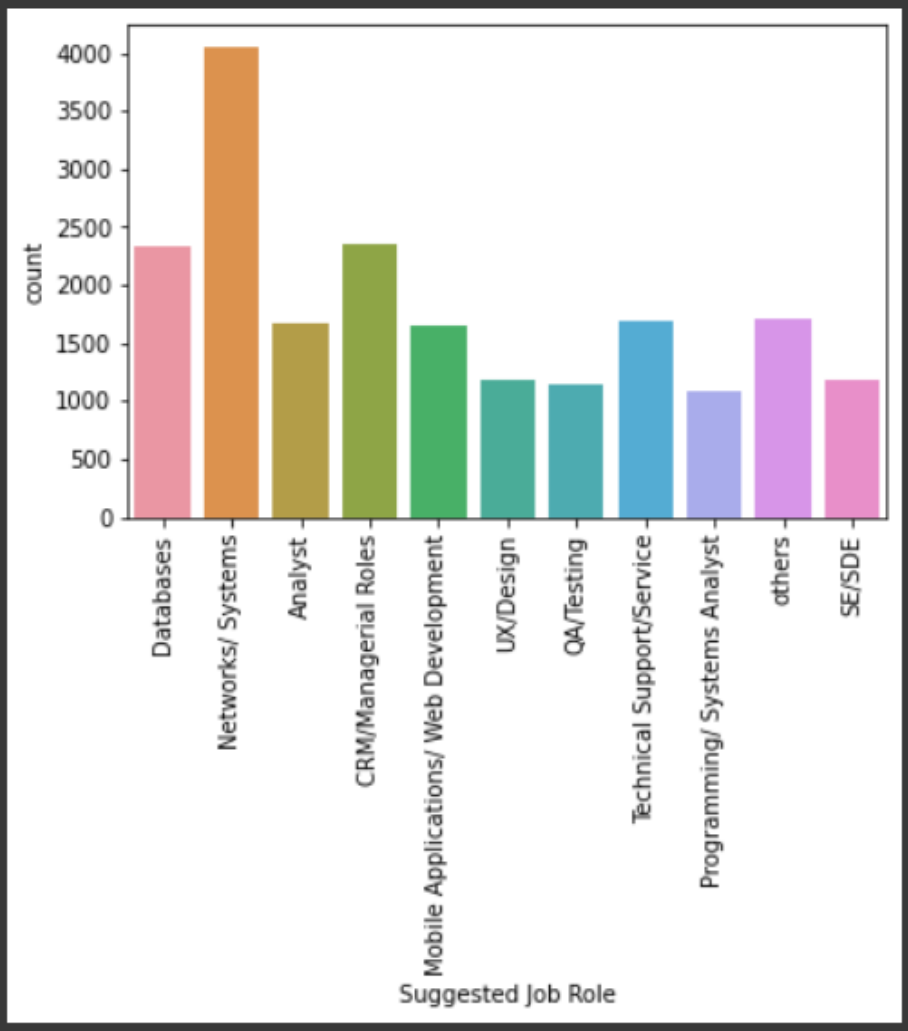
```
Train confusion matrix
[[368   0   0 ...   0   0   1]
 [  0 392   0 ...   0   0   0]
 [  4   2 440 ...   3   0   0]
 ...
 [  4   1   2 ... 439   0   4]
 [  2   1   0 ...   1 443   1]
 [  0   0   0 ...   2   0 427]]

Test confusion matrix
[[3 3 2 ... 3 6 1]
 [1 2 1 ... 2 5 4]
 [2 3 1 ... 6 3 2]
 ...
 [5 8 0 ... 5 3 3]
 [8 4 2 ... 6 4 2]
 [1 7 7 ... 4 3 7]]

Train classwise accuracies
[0.91770574 0.9468599  0.90534979 0.90380313 0.89777778 0.88864629
 0.92735043 0.94954128 0.95424837 0.93803419 0.92410714 0.94050343
 0.89548694 0.92190889 0.93706294 0.91809524 0.95305677 0.92857143
 0.93576017 0.86036036 0.95901639 0.96832579 0.91313559 0.96280088
 0.95111111 0.95116773 0.93205945 0.93607306 0.90809628 0.91898148
 0.96543779 0.8815261  0.90778689 0.92224622]

Test classwise accuracies
[0.02941176 0.02409639 0.00884956 0.01785714 0.01709402 0.03409091
 0.03278689 0.02479339 0.01834862 0.04477612 0.01680672 0.04347826
 0.01086957 0.04958678 0.03305785 0.03061224 0.07287449 0.02325581
 0.03389831 0.02040816 0.03225806 0.00724638 0.01769912 0.03636364
 0.00826446 0.01769912 0.008      0.02941176 0.02325581 0.04444444
 0.02702703 0.0390625  0.03030303 0.056     ]
```

Testing and Training accuracies are low. To increase this, I used StandardScalar and Normalization with clubbing and splitting the train and test to increase the accuracy.

The clubbing technique I used is:-

```
['Database Manager','Project Manager','Information Technology Manager'],'Manager')
['Solutions Architect','Data Architect','Information Technology Auditor','Software Quality Assurance (QA) / Testing','Quality Assurance Associate'],'others')
['Software Developer','Database Developer','Mobile Applications Developer','Web Developer','CRM Technical Developer','Applications Developer'] ,'Developer')
['Technical Engineer','Technical Services/Help Desk/Tech Support','Technical Support'],'Technical Support')
['Software Engineer','Network Security Engineer','Network Engineer','Software Systems Engineer',] ,'Engineer')
['UX Designer','Design & UX'] , 'Designer')
['Database Administrator','Portal Administrator''Network Security Administrator','Systems Security Administrator',] , 'Administrator')
['CRM Business Analyst','Programmer Analyst','Systems Analyst','Information Security Analyst','Business Systems Analyst','Business Intelligence Analyst',
 'E-Commerce Analyst'],'Analyst')
```

```
chart = sns.countplot(x=last_trans)
chart.set_xticklabels(chart.get_xticklabels(),rotation=90)
plt.show()
```



After this, the net accuracy was 0.1455.